Sudan University of Science and Technology

College of Graduate Studies

The Effect of Compression on Audio Steganography

تأثير خوارزميات الضغط على إخفاء الصوت

Thesis Submitted in Partial Fulfillment of the Requirement for Degree of Master in Computer Science

Prepared by:

Hala Mustafa Abdalraheem

Supervised by

Dr. Talaat Mohielddin Wahby

April 2017

الآيـــــــــــة

قال تعالى :

"يَوْمَئِذٍ يَتَّبِعُونَ الدَّاعِيَ لا عِوَجَ لَهُ وَخَشَعَتِ الأَصْوَاتُ لِلرَّحْمَنِ فَلا تَسْمَعُ إِلا هَمْسًا"

صدق الله العظيم

سورة طه (108)

ACKNOWLEDGEMENT

First and last thanks to **ALLAH**

All the regards and respect to the light of the dark roads we did across:

DR. Talaat Mohielddin Wahby

Rayan Kamal for his continues help in my project.

for everyone knows my names, thank you

DEDICATION

First and last Praise is to Allah
This thesis work is dedicated to the soul of all my life, my lovely **Mother**, to my **Father**, my first teacher,

This work is also dedicated to my husband, **Rayan Kamal**, who has been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having you in my life

My baby (**Dima Rayan**). You have made me stronger, better and more fulfilled than I could have ever imagined. I love you to the moon and back.

To the source of my happiness **brothers**, and **sisters**,

To the taste of the most beautiful moments with my **friends**
To the people who paved me way of science and knowledge
All my teachers

Abstract

Steganography is the hiding for important and different structure of information in other medium without discovered method by unauthorized.

In this research, a system have been proposed to enhance data hiding scheme using multilevel audio steganography and lossless compression. Multi-level steganography have been applied to increase the security of the system, in first level Enhanced Random Least Significant Bits (LSB) technique has been used and in second level Discrete Wavelet Transform (DWT) has been used.

Lossless data compression technique Huffman and LZW between first and second level of steganography were applied. Comparative analysis of proposed method has been done on basis of parameters like PSNR, MSE. The results showed that the compression algorithms have no significant impact on the process of steganography. Also, results demonstrated that the Huffman compression algorithm highly efficient than LZW comparison algorithm. Comparison was based on the PSNR values that were obtained, as an example, the value of PSNR is equal to 81.95 dB when using Huffman to compress first audio file and the value is 77.10 dB when using LZW algorithm for audio compression.

الخلاصة

الإخفاء (Steganography) هو اخفاء المعلومات المهمة والمختلفة الهيئات داخل وسائط أخرى بطريقة لاتسمح للمتطفل بإكتشافها. في هذا البحث، اقترحنا طريقة إخفاء البيانات في ملفات الصوت بإستخدام الإخفاء متعدد المستويات. في المستوى الأول تم استخدام خوارزمية (LSB) لإخفاء النص داخل الصوت، وتم إستخدام خوارزمية (DWT) في المستوى الثاني لإخفاء ملف الصوت داخل صورة. تم ضغط ملف الصوت بإستخدام خوارزمية (Huffman) وفي المرة الثانية بإستخدام خوارزمية (LZW). عملية تحليل البيانات تمت بناءاً على حساب عاملين اساسين هما (PSNR) وال(MSE). النتائج المستخلصة اظهرت ان خوارزميتي الضغط ليستا ذات تأثير كبير على عملية الإخفاء. كما أثبتت النتائج ان خوارزمية الضغط (Huffman) ذات كفاءة عالية مقارنة بخوارزمية (LZW). تمت المقارنة بناءاً على القيم الخاصة بال(PSNR) التي تم استخلاصها من النتائج، في ملف الصوت الأول على سبيل المثال كانت قيمة الـ(PSNR) لخوارزمية (Huffman) 81.95 ديسبيل بينما كانت قيمته عند استخدام خوارزمية (LZW) 77.10 ديسبيل.

Table of Contents

List of Figures

List of Tables

CHAPTER I

INTRODUCTION

1.1 Introduction

Information security is the process of protecting the availability, privacy, and integrity of data. While the term often describes measures and methods of increasing computer security, it also refers to the protection of any type of important data [1].Cryptography, and Steganography can be used in information security

Cryptography is the ability to send information between participants, in a mangled format, that prevents others from reading it. The proposed method of information security in the research is steganography.

Steganography is the art and science of hiding secret information in a cover file such that only sender and receiver can detect the existence of the secret information, Steganography can be classified into image, text, audio and video, based on the cover media used to embed secret data. The goal of steganography to hide message in such a way that no one apart from the intended recipient even know that the massage has been sent. [1]

Any steganography technique has to satisfy two basic requirements. The first requirement is perceptual transparency, cover object (object not containing any additional data) and stego-object (object containing secret message) must be perceptually indiscernible. The second constraint is high data rate of the embedded data. All the stego-applications, besides requiring a high bit rate of the embedded data, have need of algorithms that detect and decode hidden bits without access to the original multimedia sequence.

Multi-Level Steganography (MLS), which defines a new concept for hidden communication in telecommunication networks. In MLS, at least two stenographic methods are utilized simultaneously, in such a way that one method (called the upper-level) serves as a carrier for the second one (called the lower-level). Such a relationship between two (or more) information-hiding solutions has several potential benefits. The most important is that the lower-level method's stenographic bandwidth can be utilized to make the secret data unreadable even after the detection of the upper-level method. [1]

Audio steganography is embedding secret message into digital sound.

There are several techniques are available for audio steganography.

Some of them are as follows:

A. Least Significant Bit [LSB] Technique

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file. By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded.

B. Phase Coding

Phase coding addresses the disadvantages of the noise inducing methods of audio steganography. Phase coding relies on the fact that the phase components of sound are not as perceptible to the human ear as noise is. Rather than introducing perturbations, the technique encodes the message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to perceived noise ratio.

C. Echo Hiding

In echo hiding, information is embedded in a sound file by introducing an echo into the discrete signal. Like the spread spectrum method, it too provides advantages in that it allows for a high data transmission rate and provides superior robustness when compared to the noise inducing methods.

D. Spread Spectrum

In the context of audio steganography, the basic spread spectrum (SS) method attempts to spread secret information across the audio signal's frequency spectrum as much as possible. This is analogous to a system using an implementation of the LSB coding that randomly spreads the message bits over the entire sound file.

E. Discrete wavelet transform

Wavelets are small waveforms. Wavelet transform is to change the form of signal to visualize it in some different format. Discrete wavelet transform separates the data into various frequency components. [2]

This transformation of signal allows us to separate the frequency component at specific time from the other components. [2]

1.2 Problem Statement

Using on level of steganography can result in a vulnerable system, to add another layer of security two levels of steganography will be used. The proposed system uses Enhanced LSB as first steganography level, then uses DWT as second level steganography

Lossless compression algorithms (LZW and Huffman) will be used in this system and study the effect of these algorithms on audio steganography.

.

1.3 Objective of the Research

- Apply steganography to conceal secret data into the audio file.
- Enhancing Confidentiality by using two-level audio steganography.
- Apply compression on multilevel audio steganography.
- Enhance Transmission by reducing the size of the audio file.
- Enhance capacity of transmission by reducing the size of the audio file.
- Adding complexity to the system by using two-levels of audio steganography.

1.4 Methodology

We will first apply two levels of audio steganography. The first level will use enhanced Random bit LSB encoding technique is simple and efficient. The proposed technique does not affect the quality of the stego audio signal and is more secure than the conventional LSB techniques. This random bit technique do not affect the audio quality of Stego audio file and the embedded secret information can be recovered with the help of reverse steganography algorithm.

The second level will use discrete wavelet transform (DWT) which hides the audio file along the frequency distribution of the carrier image cover.

For the compression we will use two lossless methods, The First Method Huffman Coding which is based on the frequency of occurrence of a data item. The technique of Huffman Coding is to use a lower number of bits to encode the data in to binary codes that occurs more frequently.

The Second Method LZW (Lempel-- Ziv--Welch) compression is the most popular method used for the compression. The main steps for this technique is that it will read the file and given a code to each character, if the same characters are found in a file then it will not assign the new code and then use the existing code from a dictionary. The process is continuous until the characters in a file are null.

For Comparative analysis of multilevel audio we will use two parameters, First is Peak signal to noise ratio (PSNR) is used to compute how well the methods perform. PSNR compare the original image with stego image with audio file inside.
Second is Mean Squared Error (MSE) it measures the distortion in the image. It defines the square of error between original image and stego image with audio file inside.

1.5 Research Organization

Chapter one gives introduction and brief history about the steganography, defining the types of steganography and multilevel steganography. Recently literatures review and related works will be explained in chapter two. Chapter three explains the proposed algorithm, tools and techniques used in the project. The analysis of the proposed algorithm and discussion of the results appears in chapter four and finally Chapter five contains the conclusion, recommendations and future work.

CHAPTER II

LITERATURE REVIEW AND RELATED WORK

2.**1 Introduction**

Information hiding has recently gained importance in various applications. Cryptography, and Steganography can be used in information security.

Cryptography is the process of conversion of data into scrambled code that can be deciphered and sent across a public or private network. The two main forms of encrypting data in cryptography are symmetrical and asymmetrical. Symmetric encryptions, or algorithms, use the same key for encryption as they for decryption [3] other names for this type of encryption are secret-key, shared-key, and private key. Symmetric cryptography is at times simple to decode [4].

Asymmetric cryptography uses different encryption keys for encryption and decryption.

Steganography is the technique to hide the information in some media (cover media) so that third party or attacker can’t recognize that information is hidden into the cover media. The information that to be hidden is called stego and the media in which the information is hidden is called host. Various files can be act as a cover media like text, image, audio, video, IP Datagram etc. The main approach of steganography is to make difficult data discovery. [5]

Steganography and Cryptography are both used in transfer secure data to ensure data confidentiality. However the main difference between them Cryptography deals message encryption but the communication is easily aroused suspicious but on the other hand, steganography deals with secret message hiding but the communication is invisible. This is the major differences between cryptography and steganography.

It is often thought that by encrypting the traffic, the communications will be secured but this has not been adequate in real live situation [6] in cryptography method, people become aware of the existence of information by observing coded information, although they are unable to comprehend the information. Steganography hides the existence of the message so that intruders can’t detect the communication and thus provides a higher level of security than cryptography.

This makes steganography one of the popular approaches for information hiding.

The steganography application hides different types of data within a cover file. The resulting stego also contains hidden information, although it is virtually identical to the cover file.

What Steganography essentially does is exploit human perception; human senses are not trained to look for files that have information hidden inside of them, although there are programs available that can do what is called Steganalysis (Detecting use of Steganography) [7]

The below table show the main differences between steganography and cryptography:

Criterion/Method	**Steganography**	**Cryptography**
Carrier	Any digital media	Usually text based, with some extensions to image files
Key	Optional	Necessary
Detection	Blind	Blind
Authentication	full retrieval of data	Full retrieval of data
Objective	Secret communication	Data protection
Result	Stego-file	Cipher-text
Concern	Delectability/ capacity	Robustness
Type of attacks	Steganalysis	Cryptanalysis
Visibility	Never	Always

Table 2.1: the main differences between steganography and cryptography [7]

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy.

Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display [8]. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. [9] Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding.

Figure 2.1 shows the four main categories of file formats that can be used for steganography.

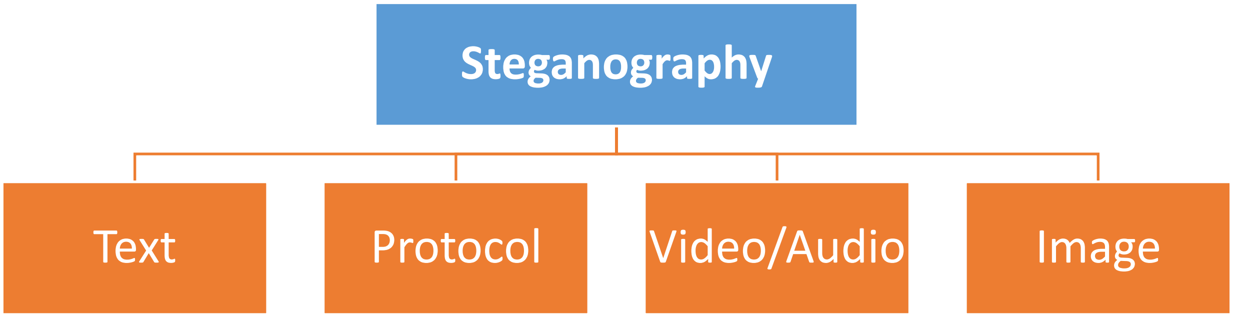


Figure 2.1: Categories of steganography

The basic terminologies used in steganography systems are: the cover message, secret message, the secret key and embedding algorithm**.** In this research work, the embedding algorithm is the compression algorithm. The cover message is the carrier of the message such as image, video, audio, text or some other digital media. Here the carrier is an audio. The secret message is the information which is needed to be hidden in the suitable digital media. The secret information in this work is in the form of any text format. The secret key is usually used to encrypt the message to have more security. [10]

2.2 Audio Steganography

In a computer-based audio steganography system, secret messages are embedded in digital sound. The secret message is embedded by slightly altering the binary sequence of a sound file. Existing audio steganography software can embed messages in WAV, AU, and even MP3 sound files [11].

Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media, such as digital images. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced. These methods range from rather simple algorithms that insert information in the exploit sophisticated signal processing techniques to hide information [11].

Signals are processed either on LSB, Parity, Phase, spread spectrum, Fast Fourier transform methods. Audio has been taken as the carrier to send the encoded messages using the above said methods. In this research, we will first apply two levels of audio steganography. The first level will use enhanced random Least Significant Bit (LSB) steganography the second level will use discrete wavelet Transform (DWT).

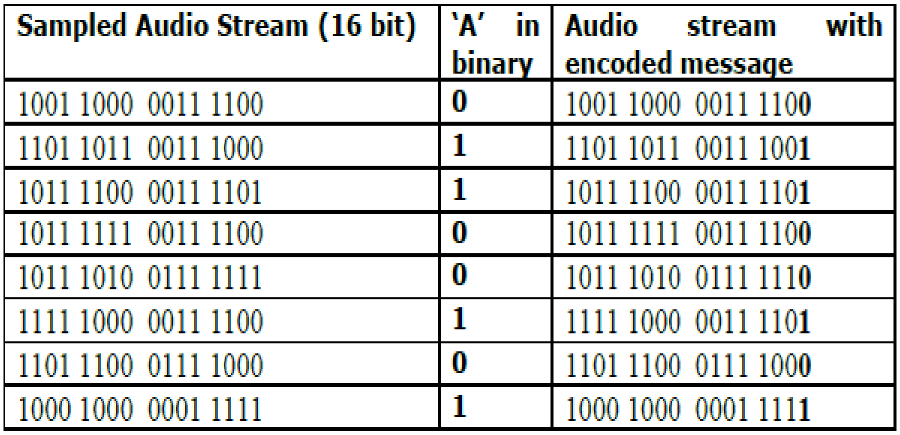
2.2.1 Least Significant Bit (LSB) Encoding

Least significant bit (LSB) coding is the simplest way to embed information in a digital audio file. By substituting the least significant bit of each sampling point with a binary message, LSB coding allows for a large amount of data to be encoded. In LSB coding, the ideal data transmission rate is 1 kbps per1 kHz. In some implementations of LSB coding, however, the two least significant bits of a sample are replaced with two message bits. This increases the amount of data that can be encoded but also increases the amount of resulting noise in the audio file as well. [12]

To extract a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. [12]

There are two main disadvantages associated with the use of methods like LSB coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, Second disadvantage it has considerably low robustness against attacks. [13]

Example:



Sampled Audio Stream (16 bit)	'A' in binary	Audio stream with encoded message
1001 1000 0011 1100	0	1001 1000 0011 1100
1101 1011 0011 1000	1	1101 1011 0011 1001
1011 1100 0011 1101	1	1011 1100 0011 1101
1011 1111 0011 1100	0	1011 1111 0011 1100
1011 1010 0111 1111	0	1011 1010 0111 1110
1111 1000 0011 1100	1	1111 1000 0011 1101
1101 1100 0111 1000	0	1101 1100 0111 1000
1000 1000 0001 1111	1	1000 1000 0001 1111

Figure 2.2: Example of LSB encoding.

2.2.2 Discrete wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) is any wavelet transform which is discretely sampled which captures both frequency and location information. A wavelet transform splits into basis functions which are known as wavelets. [2]

In this method a steganography technique to hide an audio sample in a cover image. Image can be in any format like .jpg, .bmp etc. and audio also can be in any format like .wav, .mp3 etc. Since audio files contain large no. of samples even for small duration, the cover image has to be considerably large. Color images are suitable because of enough hiding space. [2]

2.3 Multi-Level Steganography

A new concept for hidden communication in telecommunication networks. In MLS, at least two stenographic methods are utilized simultaneously, in such a way that one method (called the upper-level) serves as a carrier for the second one (called the lower-level). Such relationship between two (or more) information hiding solutions has several potential benefits. The most important is that the lower-level method stenographic bandwidth can be utilized to make the steganogram unreadable even after the detection of the upper-level method: e.g., it can carry a cryptographic key that deciphers the steganogram carried by the upper-level one. It can also be used to

Provide the steganogram with integrity. Another important benefit is that the lower-layer method may be used as a signaling channel in which to ex-change information that affects the way that the upper-level method functions, thus possibly making the steganography communication harder to detect.[1]

2.4 Compression

Compression is the conversion of data in such a format that requires few bits usually formed to store and transmit the data easily and efficiently. Compression is used to reduce amount of data and needed to reproduce that data whenever we require it. [12]

Compression is the combination of two components. One is encoding algorithm, another one is decoding algorithm. In encoding algorithm makes the message as compressed representation. The decoding algorithm reconstructs the message from compressed representation to original message or it reconstructs some approximation.

There are two type of compression methods – lossy compression and lossless compression. In lossy compression some data loss may occurs after compression while in lossless compression no such data loss occurs. Steganography completely deals with data security hence lossless compression techniques are more preferable. [12]

2.4.1 Lossless Compression

Lossless compression techniques, as their name implies, involve no loss of information. If data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data.

Text compression is an important area for lossless compression. It is very important that the reconstruction is identical to the text original, as very small differences can result in statements with very different meanings. [14]

Lossless compression researchers have developed highly sophisticated approaches, such as Huffman encoding, arithmetic encoding the Lempel-Ziv (LZ) family, Dynamic Markov Compression (DMC), Prediction by Partial Matching (PPM), Run length coding (RLE) and Burrows-Wheeler Transform (BWT) based algorithms. [14]

However, none of these methods has been able to reach the theoretical best-case compression ratio consistently.

In this research we will use two lossless methods, the first method Huffman Coding the Second Method LZW (Lempel--Ziv--Welch).

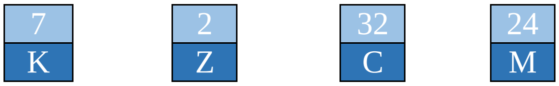
2.4.1.1 Huffman Coding

The Huffman coding algorithm works on bottom-up approach is named after its inventor, David Huffman, who developed the method as a student in a class on information theory at MIT in 1950[12]. Huffman coding is based on frequency of occurrence of a data item.

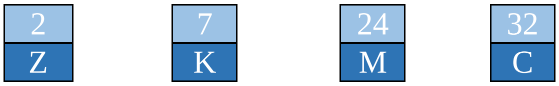
Huffman algorithms have two ranges static as well as adaptive. Static Huffman algorithm is a technique that encodes the data in two passes. In first pass, it is required to calculate the frequency of each symbol and in the second pass it constructs the Huffman tree. Adaptive Huffman algorithm is expanded on Huffman algorithm that constructs the Huffman tree in one pass but takes more space than Static Huffman algorithm.

Huffman coding creates the tree like structure when it encodes the given string. The example of Huffman Encoding with algorithm is as follows [15].

Step 1: Input String

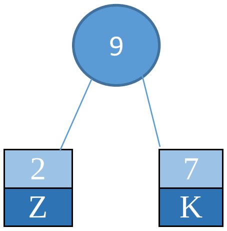


Step 2: Sorting date by frequencies



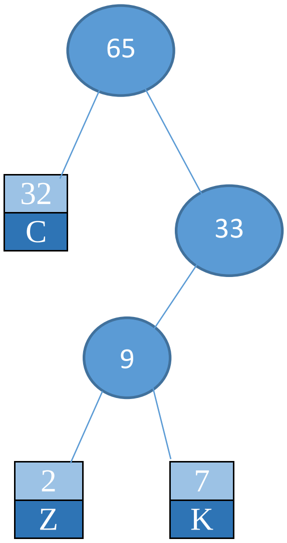
Step 3: Choose two smallest frequencies count

Step4: Merge them together with the sum of their frequencies and update the data



Step 5: Repeat steps 2-4.

The final Huffman tree will be:



2.4.1.2 Lempel–Ziv–Welch (LZW)

LZW (Lempel-Ziv–Welch) is a totally dictionary based coding. Lzw encoding is further divided into static & dynamic. In static, dictionary is fixed during the encoding and decoding processes. In dynamic dictionary coding, the dictionary is updated if needed LZW compression replaces strings of characters with single codes. It does not perform any analysis of the incoming text. Instead, it just adds every new string of characters from the table of strings. The code that the LZW algorithm outputs can be of any arbitrary length, but it must have more bits in it than a single character. LZW compression works best for files containing lots of repetitive data. LZW compression maintains a dictionary. In this dictionary all the stream entry and code are stored. [15]

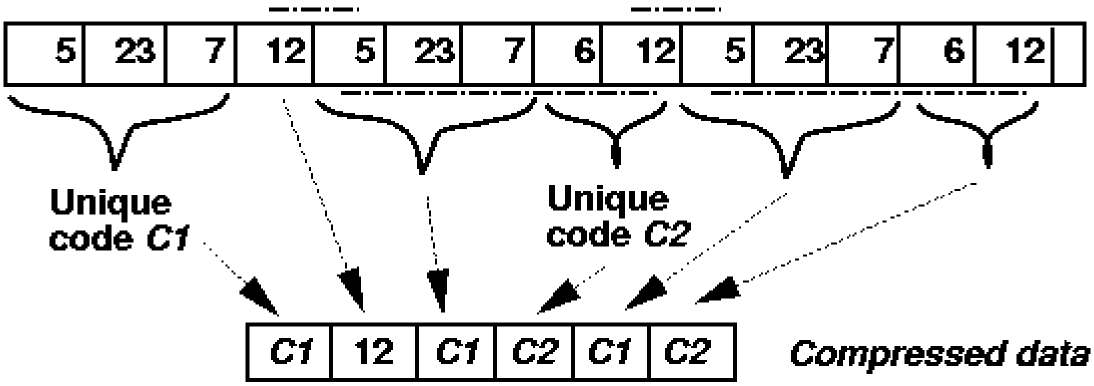


Figure 2.3: Example of LZW coding.

The basic steps of LZW algorithm are as follows:

Step 1: Input the stream.

Step 2: Initialize the dictionary to contain entry of each character of stream.

Step 3: Read the stream if current byte is end of the stream, then exit.

Step 4: Otherwise read next character and produce a new code.

If the bunch of character is frequently occurring then give them a unique code (according to the diagram)

Step 5: Read next input character of stream from dictionary if there is no such character in dictionary, then

A: Add new string to the dictionary.

B: Write the new code for new entered string.

C: Go to step 4.

Step 6: Write out code for encoded string and exit [15]

2.5 Related works

Section two in this chapter presents the related work in audio steganography and present compression table between them

[12] Chintan R. Nagrecha,Prof. Prashant B. Swadas

This paper deals with the approach of embedding the bits at higher random layer which leads towards difficult discovery of data. Main aim of this paper is to improve capacity and robustness of this approach.

Step 1: Extract host audio file, evaluate sample rate, sample size etc. according to sample size (16bit or 8bit) read sample.

Step 2: Read message string.

Step 3: According to message length choose compressive algorithm (i.e., for extremely large message prefer Shannon- Fano) and embed output bit stream over audio.

Step 4: Generate new 16 bit samples by inserting message bits into random higher bits using algorithm.

Step 5: Embedding message bit at random higher layers such that the distortion can be minimized.

Step 6: Embed layer number (bit position) value with next sample.

Step 7: Convert all sample (16 bit) into regular audio stego file.

In standard LSB algorithm the chances of message discovery is very high. Difficult discovery of message bit in host audio can be achieved by embedding it to random higher bits of sample. After applying compression algorithm the output message bit stream completely differs then input message. As compare to standard method this approach is more robust. Lossless compression techniques sufficiently improve storage capacity of host audio.

[16] AmbiKadev, LijilDomininc, Swetha

This paper proposes and analyzes a methodology for video steganography that pre-processes the text before hiding it behind a cover video. In pre-processing process, the text is first compressed and then modified using a key. The proposed method combines the idea of video steganography, cryptography and compression techniques which provide enhanced security. The RC4 is used for encrypting the compressed text and 4LSB method is used to conceal the processed text inside the cover media, while the purpose of compression is to enhance security and increase embedding capacity.

The proposed technique for steganography is named as Compress-Encrypt-Steganography. This method pre-processes the text before hiding it behind a cover image. In the proposed method AVI video is used as the cover medium. Each frame will be an image. In the sender side, the proposed method involves 3 stages namely frame conversion, preprocessing and embedding. In frame conversion phase as video is a bunch of frames, for easy processing the video is converted into component frames. In pre-processing process, the text is first compressed using LZW, which is a dictionary method of compression. The compression is followed by encryption.RC4; a stream cipher method is used for encrypting the compressed text. The next stage is embedding, where data is concealed behind cover image using 4LSB method. Whether data is hidden or not is represented in the first frame of the video. Using 4LSB method, the key for the encryption is hidden in the |_n/3_|th frame and the preprocessed text is hidden in the |_n/2_|th frame. After embedding process the frames are recombined to form the stego video. The stego video is then passed through the communication channel. In the receiver side, the proposed method involves 3 stages namely frame conversion, undo stage and extraction. On receiving the stego video, initially the stego video is converted into frames in the frame conversion stage. The frames are then forwarded to extraction stage. In the extraction phase, using reverse 4LSB algorithm the key is extracted from the |_n/3_|th frame and the preprocessed text from the |_n/2_|th frame. The key and the preprocessed text is the moved to undo stage. In undo stage the preprocessed text is first decrypted using the key and decompressed. The output of the undo stage will be the secret message.

Cascading multiple algorithms increased time complexity but in turn the system provides enhanced security and embedding capacity.

[10] Tamanna Garg, Sonia Vatta

The main goal of this research work is to enhance the data compression rate by designing and applying compression algorithm on bmp images to facilitate the hiding of enlarged text in an image.

Compression algorithm which has been used to design an application capable of hiding large documents in small images without any changes. The developed compression algorithm has a capability of hiding data up to eight bits per pixel.

In this research work, the embedding algorithm is the compression algorithm. The cover message is an image. The secret information in this work is in the form of any text format. It uses LSB substitution for embedding the data into images to enhance the security of the secret message it used the covering of resulting stego image with a new cover image The new cover image can be the same or different than the original. In order to increase the storage capacity of the image, a compression algorithm has been used.

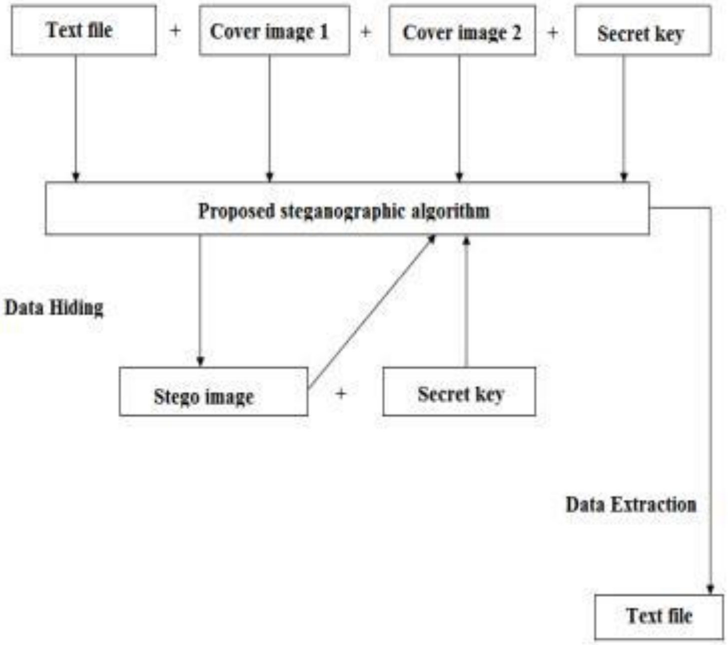


Figure 2.4: General layout of proposed system. [16]

[17] Rahul Jain , Naresh Kumar.

In this work it explored the existing image steganography techniques. It proposed an efficient image Steganography technique. In image steganography, image is used as a carrier for transmission of the secret information or data. The image used can be either gray scale or color image. In this technique data is firstly preprocess. This preprocessing reduces the size of the data by a significantly great amount. This preprocessed data is then embedded into the LSBs of the pixels of the image depending upon the intensity of the pixel values. For pre-processing a lossless data compression technique, LZW (Lempel–Ziv–Welch) technique is used. In this technique sequence of 8-bit secret data is encoded as fixed-length 12-bit codes. The code from value 0 to 255 represents one character sequences consisting of the corresponding 8-bit

Character. As the data is encoded, the codes with values 256 through 4095 are created in a dictionary depending upon the sequences encountered in the data .A dictionary is initialized to contain the single-character strings corresponding to all the possible input characters. At every step in the compression process, input characters are gathered into a sequence until the next character comes that will make a sequence for which there is no code in the dictionary.

Steganography technique used is Modified Kekre Algorithm (MKA). In this technique, firstly 8-bit secret bit is selected. The secret bit is XORed with all the bytes of the secret message that is to be embedded into the cover image .For the pixels of the cover image having intensity value greater than 239; if the bit 1 is to be embedded then 5 bits of secret text are embedded and for embedding bit 0, 4 bits are embedded in LSBs of that pixel. For a pixel having intensity from 224 to 239; if bit to be embedded is 0, 5 bits of the secret message are embedded and for bit 1, 3 bits are embedded. For a pixel having intensity value in the range of 192 to 223, 2 bits are embedded otherwise only one bit is embedded. This technique is targeted to achieve very high image embedding capacity into the cover image and more security of the secret data.

[18] Suma Christal Mary.

In this paper propose a new method for the real-time hiding of information used in compressed video bit streams. This method is based on the real-time hiding of information in audio steganography. A new compressed video secure steganography (CVSS) algorithm is proposed. In this algorithm, embedding and detection operations are both executed entirely in the compressed domain, with no need for the decompression process. The framework for CVSS included four function parts, the video sequence parser, the scene change detector, the secret message embedder and the video steganalysis. Compressed video sequence supported compression through the elimination of temporal, redundancies with the use of motion compensation, block quantization inside a discrete cosine transform (DCT), and Huffman run level encoding. With this compression, the video bit stream consists of variable length codes (VLCs). The cover video sequence *U* pass through the sequence parser module at first, the second module, scene change detector, can divide the sequence into several slow speed and single scene video sub-sequences. With the third module, message embedder, secret message *M* is hidden into the compressed video sequence without bringing perceptive distortion. The last module, video steganalysis, is used as a checker for the message hiding security in the stego video sequence. In this experiment used several frames with same sequence number from the different correlation kind compressed video streams. The PSNR value and correlation value change magnitude of the stego-video, it means the perception quality and intra frame correlation of test video is little changed.

[14] M.Baritha Beguma ,Y.Venkataramanib

This paper proposes a method of text transformation using Dictionary based encoding and audio steganography. The text message compressed by dictionary based compression, it is hidden in the audio file.

The performance issue such as compression ratio and Bits per Character (BPC) are compared for the five cases simple Arithmetic coding, Huffman with BWT, LZSS with BWT and Dictionary based Encoding (DBE) The results of Dictionary based Encoding (DBE) algorithm achieves good compression ratio, reduces bits per character.

#	Paper Name	Author	Year	Objectives
10	ENHANCING DATA COMPRESSION RATE USING STEGANOGRAPHY	Tamanna Garg, Sonia Vatta	2014	- LSB substitution for embedding the data into images.
12	Audio Steganography with Various Compression Algorithms to Improve Robustness and Capacity	Chintan R. Nagrecha* Prof. Prashant B. Swadas	2014	-Embedding message bit at random higher level.
14	LSB Based Audio Steganography Based On Text Compression	M.Baritha Beguma ,Y.Venkataramanib	2011	-Text hidden in audio steganography.
16	Compress-Encrypt Video Steganography	Liji L Dominic, Swetha , Ambikadevi Amma	2015	- LSB method is used to conceal the processed text inside the cover media.
17	Efficient data hiding scheme using lossless data compression and image steganography	Rahul Jain, Naresh Kumar	2012	-Modified Kekre Algorithm (MKA) for steganography.
18	Improved protection in video steganography used compressed video bit streams.	Suma Christal Mary	2010	- compressed video secure steganography (CVSS).

Table 2.2: Summary of related studies.

CHAPTER III

WORK ENVIROMENT AND PROPOSED SYSTEM ANALYSIS

3.1 Overview

This chapter describes the proposed method (lossless compression with multilevel audio steganography) and explains the diagrams that clarify the proposed method. In level one modified enhanced random bits LSB steganography, which is the simplest way to embed information in a digital audio file by read one byte from the source audio sample. After reading the selected byte we are needs to calculate the decimal value of that byte. As the decimal value of the read byte is between 0-255 so the embedding of data of the next sample depends upon the calculated decimal value.

Level two discrete wavelet Transform (DWT). Which embed audio file in image cover.

Compression algorithm used to compress file after level one audio steganography.

The programming language will used in implementation of the level two and Compression algorithm is java programming language, since it contains appropriate and more suitable methods to read from file, write in file.

MATLAB (R2010a) is also used to evaluate the results of the proposed method by calculating the PSNR and MSE of image file, the MATLAB is suitable for the evaluation because it’s a high-level technical computing language and an interactive environment for algorithm development.

3.2 Proposed Method

The proposed method is using multilevel audio steganography with lossless audio compression.

Level one is done by embedding the secret message (text) into cover audio (wav file) using enhanced random Least Significant Bit (LSB) audio steganography.

The output from level one is audio stego which work as input in two lossless audio compression algorithm (Huffman, LZW)

Level two is done by embedding the audio file with secret message after compression into another cover image using *DWT.*

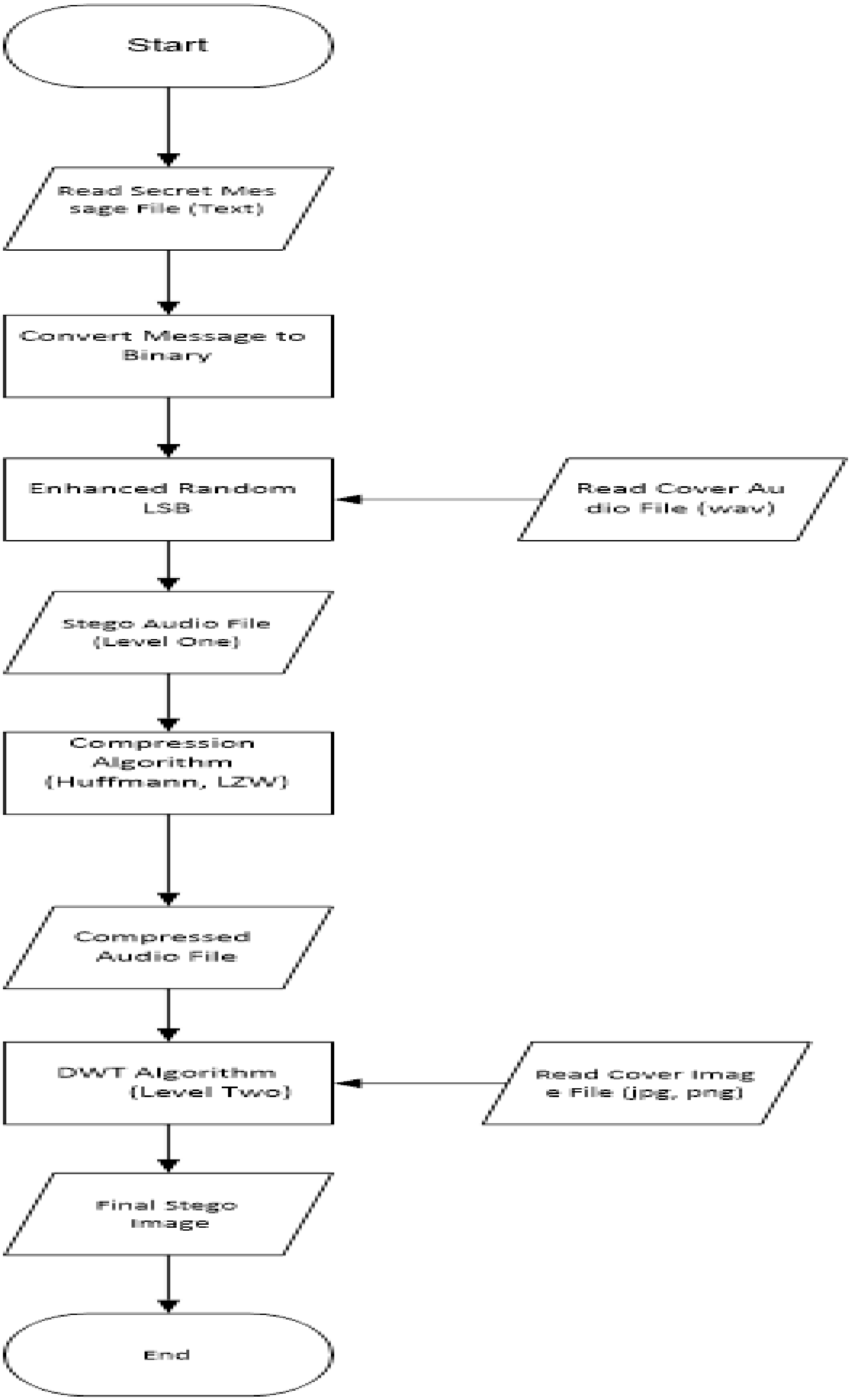

Start
Read Secret Mes sage File (Text)
Convert Message to Binary
Enhanced Random LSB
Read Cover Au dio File (wav)
Stego Audio File (Level One)
Compression Algorithm (Huffmann, LZW)
Compressed Audio File
DWT Algorithm (Level Two)
Read Cover Imag e File (jpg, png)
Final Stego Image
End

Figure 3.1: General layout of the proposed system (Embedding module)

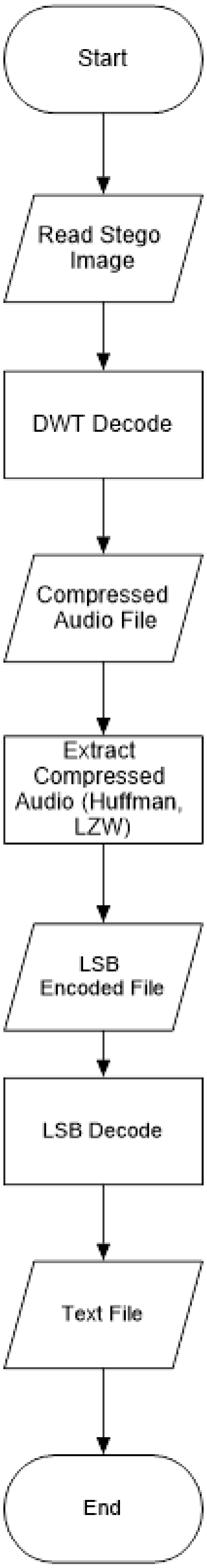


Figure 3.2: General layout of the proposed system (Extracting module)

3.2.1 Enhanced Random LSB Embedding (Level One Steganography)

1- Read audio file (.wav) as cover file.
2- Read text file.
3- Convert text file to binary.
4- Read next byte from cover audio file.
5- Calculate the decimal value of the cover audio byte.
6- Determine the bits to be encoded from message based on decimal value from step 5.
7- Embed data of the next sample according to the calculated decimal value.
8- Write new audio file after the embedding is completed.

Figure 3.3 shows the embed process for LSB.

.

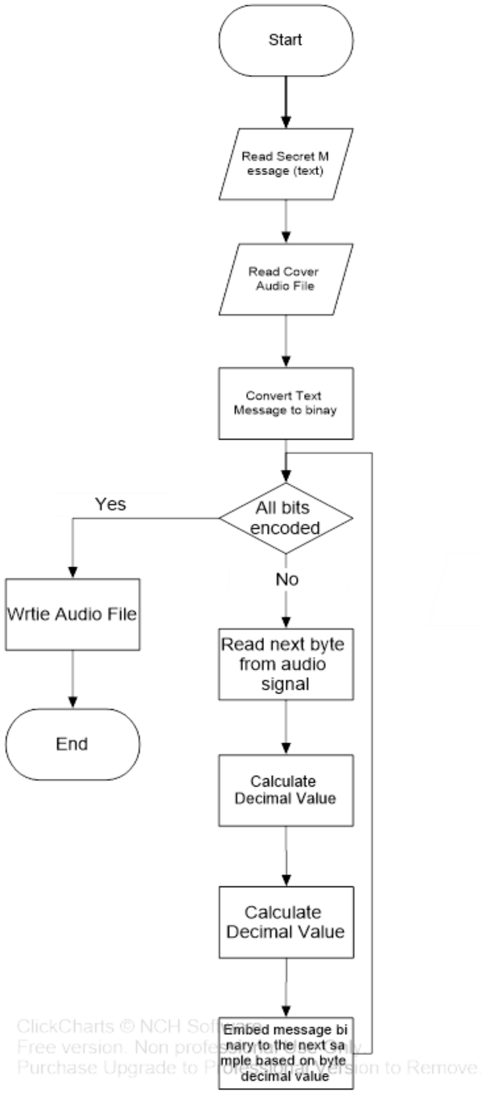

Start
Read Secret M essage (text)
Read Cover Audio File
Convert Text Message to binay
All bits encoded
Yes
No
Wrtie Audio File
End
Read next byte from audio signal
Calculate Decimal Value
Calculate Decimal Value
Embed message bi nary to the next sa mple based on byte decimal value

Figure 3.3: LSB embedding process

3.2.2 Enhanced Random LSB Extracting

1. Read LSB encoded audio file (.wav).
2. Read next byte from encoded audio file.
3. Calculate decimal value of byte from step 2.
4. Determine bits to be extracted from the byte we read from step 2.
5. Extract bits based on value from step 4.
6. Write extracted information to the text file.

Figure 3.4 shows the extract process for LSB.

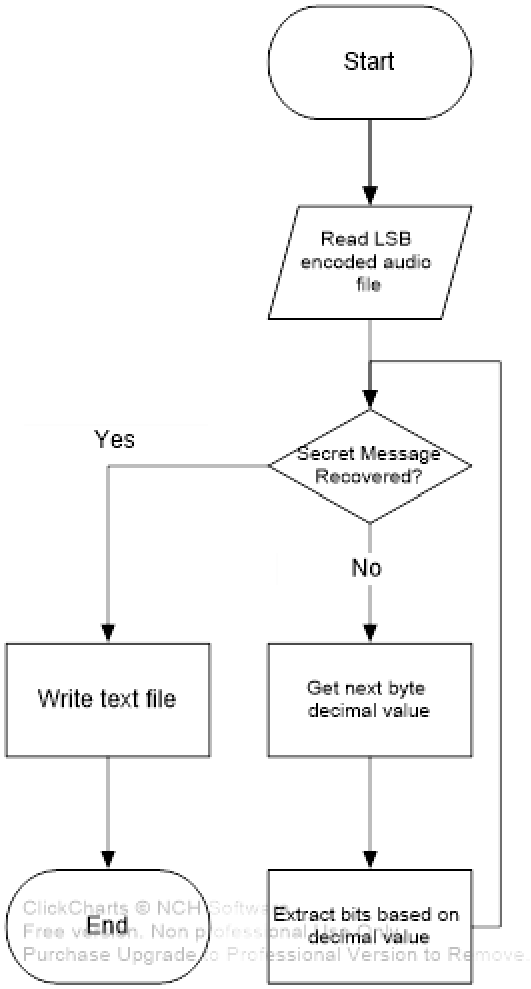

Start
Read LSB encoded audio file
Secret Message Recovered?
Yes
No
Write text file
Get next byte decimal value
End
Extract bits based on decimal value

Figure 3.4: LSB extract process.

3.2.3 Lossless Compression Algorithm Huffman

Huffman encoding takes a sequence (stream) of symbols as input and gives a sequence of bits as output. The intent is to produce a short output for the given input. Each input yields a different output, so the process can be reversed, and the output can be decoded to give back the original input.

The main steps of compression are given below:

1. Initialize Frequency table.
2. Create binary tree of nodes (stored in a regular array, the size of which depends on the number of symbols n).
3. Create a leaf node for each symbol and add it to the priority queue.
4. While there is more than one node in the queue:
    a. Remove the two nodes of highest priority (lowest probability) from the queue.
    b. Create a new internal node with these two nodes as children and with probability equal to the sum of the two nodes' probabilities.
    c. Add the new node to the queue.
    d. Assign, bit '0' represents the left child and bit '1' represents f the right child.
5. The remaining node is the root node and the tree is complete.
6. A finished tree has up to $n$leaf nodes and $n-1$internal nodes.

3.2.4 Lossless Decompression Algorithm Huffman

1. We read the coded message bit by bit. Starting from the root, we follow the bit value to traverse one edge down the tree.
2. If the current bit is 0 we move to the left child, otherwise, to the right child.
3. We repeat this process until we reach a leaf. If we reach a leaf, we will decode one character and re-start the traversal from the root.
4. Repeat this read-move procedure until the end of the message.

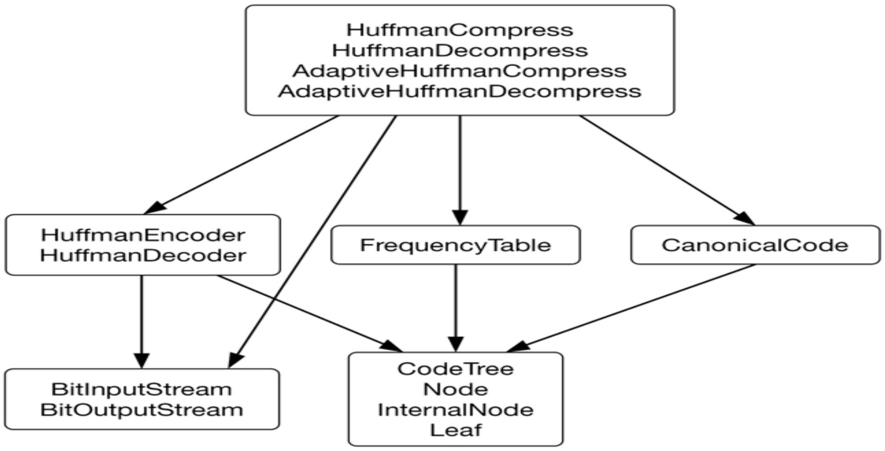


Figure 3.5: Huffman Compress/Decompress module.

3.2.5 Lossless Compression Algorithm LZW (Lempel-Ziv Welch)

Encodes sequences of 8-bit data as fixed-length 12-bit codes. The codes from 0 to 255 represent 1-character sequences consisting of the corresponding 8-bit character, and the codes 256 through 4095 are created in a dictionary for sequences encountered in the data as it is encoded. At each stage in compression, input bytes are gathered into a sequence until the next character would make a sequence for which there is no code yet in the dictionary. The code for the sequence (without that character) is added to the output, and a new code (for the sequence with that character) is added to the dictionary.

The main steps for this technique are given below:

1. First it will read the file and give a code to each character.
2. Initialize the dictionary to contain all blocks of length one.
3. If the same characters are found in a file then it will not assign the new code and then use the existing code from a dictionary.
4. Add new characters followed by the first symbol of the next block to the dictionary.
5. The process is continuous until the characters in a file are null.

LZW Compression Algorithm Code:

```
STRING = get input character
WHILE there are still input characters DO
    CHARACTER = get input character
    IF STRING+CHARACTER is in the string table then
        STRING = STRING+character
    ELSE
        output the code for STRING
        add STRING+CHARACTER to the string table
        STRING = CHARACTER
    END of IF
END of WHILE
output the code for STRING
```

3.2.6 Lossless Decompression Algorithm LZW (Lempel-Ziv Welch)

1. Reading a value from the encoded input and outputting the corresponding string from the initialized dictionary.
2. Get next code(int& code) puts the next code from the input into code
3. If the next value is unknown to the decoder, then it must be the value that will be added to the dictionary
4. Repeats the process until there is no more input, at which point the final input value is decoded without any more additions to the dictionary.

LZW Decompression Algorithm Code:

```
Read OLD_CODE
output OLD_CODE
WHILE there are still input characters DO
    Read NEW_CODE
    STRING = get translation of NEW_CODE
    output STRING
    CHARACTER = first character in STRING
    add OLD_CODE + CHARACTER to the translation table
    OLD_CODE = NEW_CODE
END of WHILE
```

3.2.7 Discrete Wavelet Transform Embedding (Level Two Steganography)

Algorithm Code:

1. Read cover image file (CI) and audio signal (AS).
2. Convert image from RGB color format to YCbCr color format using, y=rgb2ycbcr (CI).
3. Select one of the chrominance components (cb or cr) from YCbCr format using, cb=y(:,:,2) or cr=y(:,:,3)
4. Apply Haar wavelet transform on cb or cr (here assume cb is used) to get low frequency and low frequency sub-bands (LL, HL, LH, HH) by lifting wavelet using LS=liftwave(_haar‘,’int2int’) and then apply wavelet transform on cb using lwt as, [LL, HL, LH, HH]=lwt2(double(cb),LS).
5. Obtain wavelet transform of secret audio to get approximation coefficients (CA) and detailed coefficients (CD) using, [CA, CD] =lwt (double(AS),LS).
6. Hide approximation coefficients of audio (CA) in high frequency (HH) sub-band of image and detailed coefficients (CD) in another sub-band(HL).
7. Obtain inverse wavelet transform to get stego Cb. Then convert to RGB format. SCb = ilwt2 (LL, HL, LH, HH, LS) SI=ycbcr2rgb (YSCbCr) SI =imwrite(SI, _stego.jpg‘).
8. Write converted image and exit.

3.2.8 Discrete Wavelet Transform Extracting

Algorithm Code:

1. Read Stego Image (SI) and convert it from RGB to YCbCr format.

2. Extract cb component from Ey and apply haar wavelet transform to get four sub bands (ELL, EHL, ELH, EHH) Ecb=Ey(:,:,2) ELS = liftwave (_haar', 'Int2Int') [ELL, EHL, ELH, EHH] = lwt2(double(Ecb),LS).
3. Obtain inverse wavelet transform for approximation coefficients & detailed coefficients obtained in step v to get the secret audio. ESA=ilwt(ELL, EHL, EHH).
4. Write recovered audio file and exit.

Chapter IV

RESULTS AND DISCUSSION

4.1 Introduction

The proposed method is using multilevel steganography and lossless compression; level one steganography will be done by using Least Significant Bit (LSB) audio steganography. The output from level one (intermediate audio) was compressed using two types of lossless compression, Huffman coding and LZW coding. In second level using Discrete Wavelet Transform (DWT).

Comparative analysis of multilevel audio steganography (Enhanced Random LSB and DWT image steganography) has been done on basis of parameters like PSNR, MSE.

4.2 Evaluation Parameters

4.2.1 PSNR

The Peak Signal to Noise Ratio (PSNR) is the ratio between maximum possible power and corrupting noise that affect the representation of the image. PSNR is usually expressed as decibel scale. The PSNR is commonly used as measure of quality reconstruction of image. The signal in this case is the original data and the noise is the error introduced. High value of PSNR indicated the high quality of the image. The mathematical representation of the PSNR is:

$$PSNR = 20\ \log_{10}\left(\frac{\mathrm{MAX}\,f}{\sqrt{MSE}}\right)$$

4.2.2 MSE

Mean Squared Error (MSE) is the signal fidelity measure used to compare two signals by providing a quantitative score that describes the degree of similarity/fidelity or, conversely, the level of error/distortion between them.

The mathematical representation of the MSE is:

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}||\ f(i,j) - g(i,j)\ ||^{2}$$

Legend:

f represents the matrix data of the original image

g represents the matrix data of the degraded image

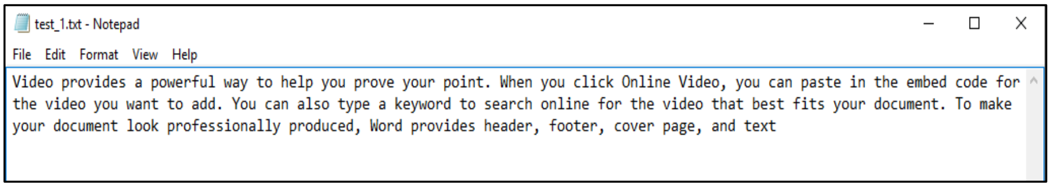
m represents the number of rows of pixels of the images and i represents the index of the row

n represents the number of columns of pixels of the image and j represents the index of that column

MAX f is the maximum signal value that exists in the original image.

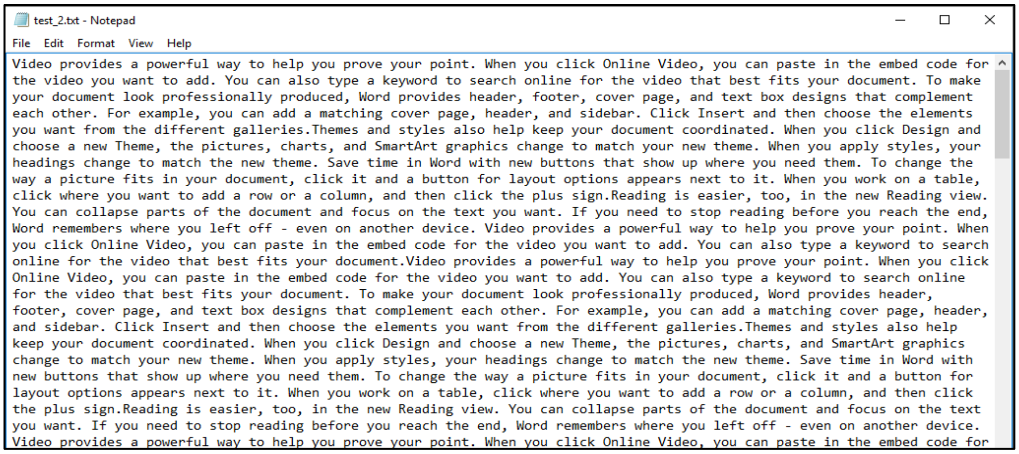
4.3 Used Secret Messages

There are three different message sizes have been used to be embedded in different audio size in the upper level of steganography, Figure 4.1, Figure 4.2 and Figure 4.3 below show example of the secret messages used.



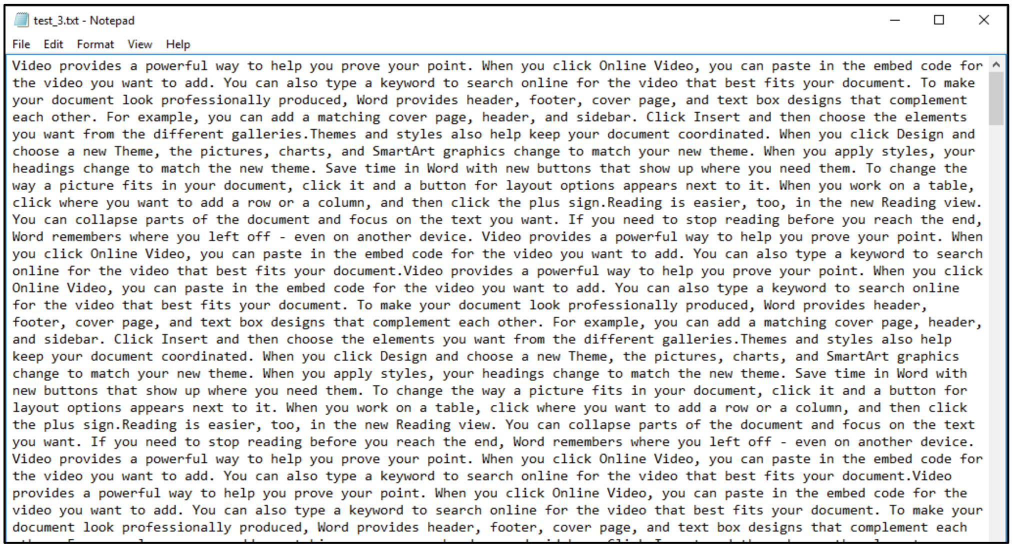
```
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for
the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make
your document look professionally produced, Word provides header, footer, cover page, and text
```

Figure 4.1: First Secret Message



```
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for
the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make
your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement
each other. For example, you can add a matching cover page, header, and sidebar. Click Insert and then choose the elements
you want from the different galleries.Themes and styles also help keep your document coordinated. When you click Design and
choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme. When you apply styles, your
headings change to match the new theme. Save time in Word with new buttons that show up where you need them. To change the
way a picture fits in your document, click it and a button for layout options appears next to it. When you work on a table,
click where you want to add a row or a column, and then click the plus sign.Reading is easier, too, in the new Reading view.
You can collapse parts of the document and focus on the text you want. If you need to stop reading before you reach the end,
Word remembers where you left off - even on another device. Video provides a powerful way to help you prove your point. When
you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search
online for the video that best fits your document.Video provides a powerful way to help you prove your point. When you click
Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online
for the video that best fits your document. To make your document look professionally produced, Word provides header,
footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header,
and sidebar. Click Insert and then choose the elements you want from the different galleries.Themes and styles also help
keep your document coordinated. When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics
change to match your new theme. When you apply styles, your headings change to match the new theme. Save time in Word with
new buttons that show up where you need them. To change the way a picture fits in your document, click it and a button for
layout options appears next to it. When you work on a table, click where you want to add a row or a column, and then click
the plus sign.Reading is easier, too, in the new Reading view. You can collapse parts of the document and focus on the text
you want. If you need to stop reading before you reach the end, Word remembers where you left off - even on another device.
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for
```

Figure 4.2: Second Secret Message #2



```
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for
the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make
your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement
each other. For example, you can add a matching cover page, header, and sidebar. Click Insert and then choose the elements
you want from the different galleries.Themes and styles also help keep your document coordinated. When you click Design and
choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme. When you apply styles, your
headings change to match the new theme. Save time in Word with new buttons that show up where you need them. To change the
way a picture fits in your document, click it and a button for layout options appears next to it. When you work on a table,
click where you want to add a row or a column, and then click the plus sign.Reading is easier, too, in the new Reading view.
You can collapse parts of the document and focus on the text you want. If you need to stop reading before you reach the end,
Word remembers where you left off - even on another device. Video provides a powerful way to help you prove your point. When
you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search
online for the video that best fits your document.Video provides a powerful way to help you prove your point. When you click
Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online
for the video that best fits your document. To make your document look professionally produced, Word provides header,
footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header,
and sidebar. Click Insert and then choose the elements you want from the different galleries.Themes and styles also help
keep your document coordinated. When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics
change to match your new theme. When you apply styles, your headings change to match the new theme. Save time in Word with
new buttons that show up where you need them. To change the way a picture fits in your document, click it and a button for
layout options appears next to it. When you work on a table, click where you want to add a row or a column, and then click
the plus sign.Reading is easier, too, in the new Reading view. You can collapse parts of the document and focus on the text
you want. If you need to stop reading before you reach the end, Word remembers where you left off - even on another device.
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for
the video you want to add. You can also type a keyword to search online for the video that best fits your document.Video
provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the
video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your
document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each
```

Figure 4.3: Third Secret Message #3

The sizes of the secret messages used is shown in table 4.1 below

Secret Message File	**Secret Message File Size (in Bits)**
First Secret Message	7,422
Second Secret Message	163,840
Third Secret Message	286,720

Table 4.1: Secret Message File Sizes

4.4 Experimental Results

After the upper level (level one Enhanced Random LSB) is applied to the messages shown on table 4.1 above, the output was three audio files. Each audio file concealing one of the secret messages. The first cover audio file is the Beep wave file and it concealed the first secret message, the size of the audio file was 11,659 bits. Figure 4.4 shows the sound file wave.

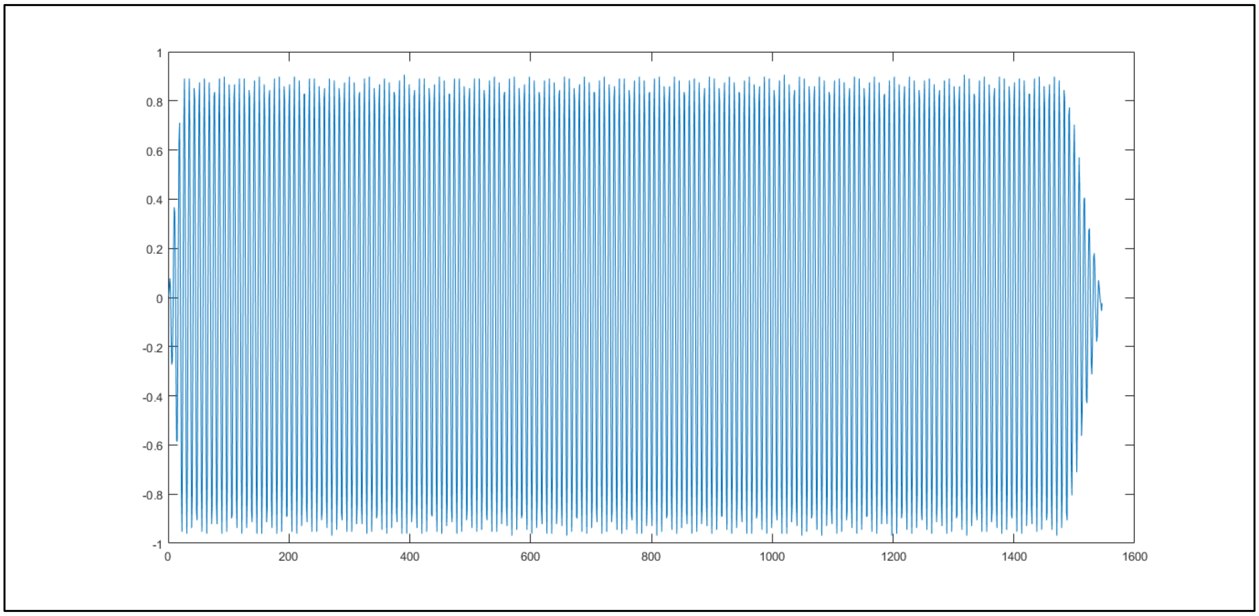


Figure 4.4: Beep.wav sound file wave

The second cover audio file was Squeeze wave file and it concealed the second secret message, the size of the audio file was 516,096 bits. Figure 4.5 shows Squeeze sound file wave.

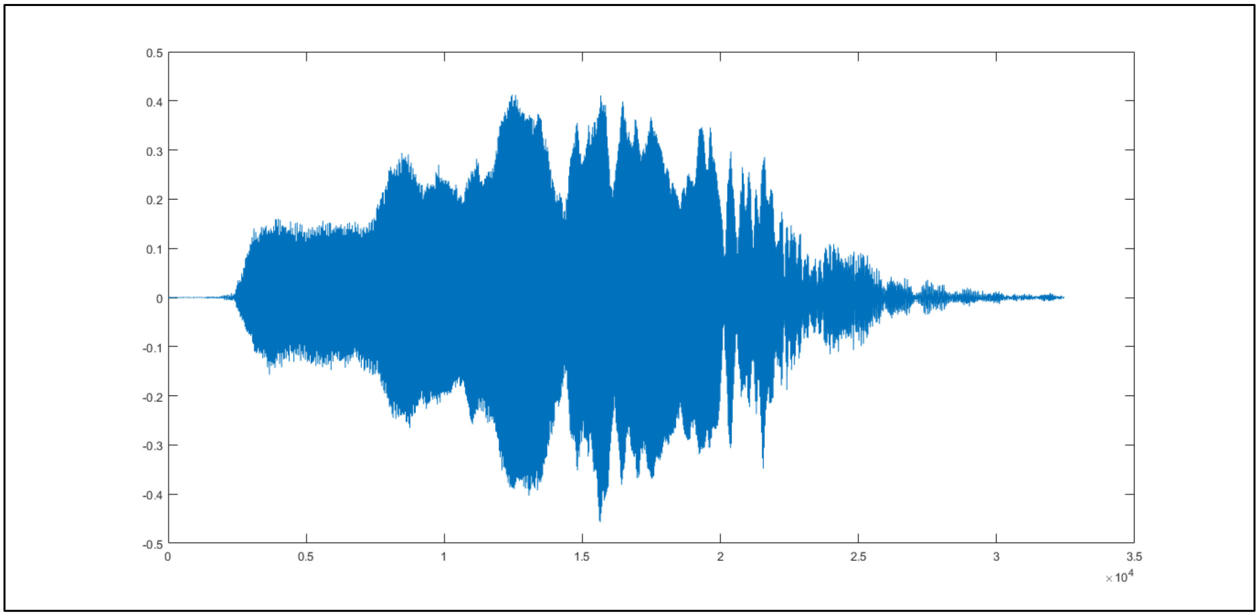


Figure 4.5: Squeeze.wav sound file wave

The third cover audio file was Chanting wave file and it concealed the third secret message, the size of the audio file was 1,163,264 bits. Figure 4.6 show the sound file wave.

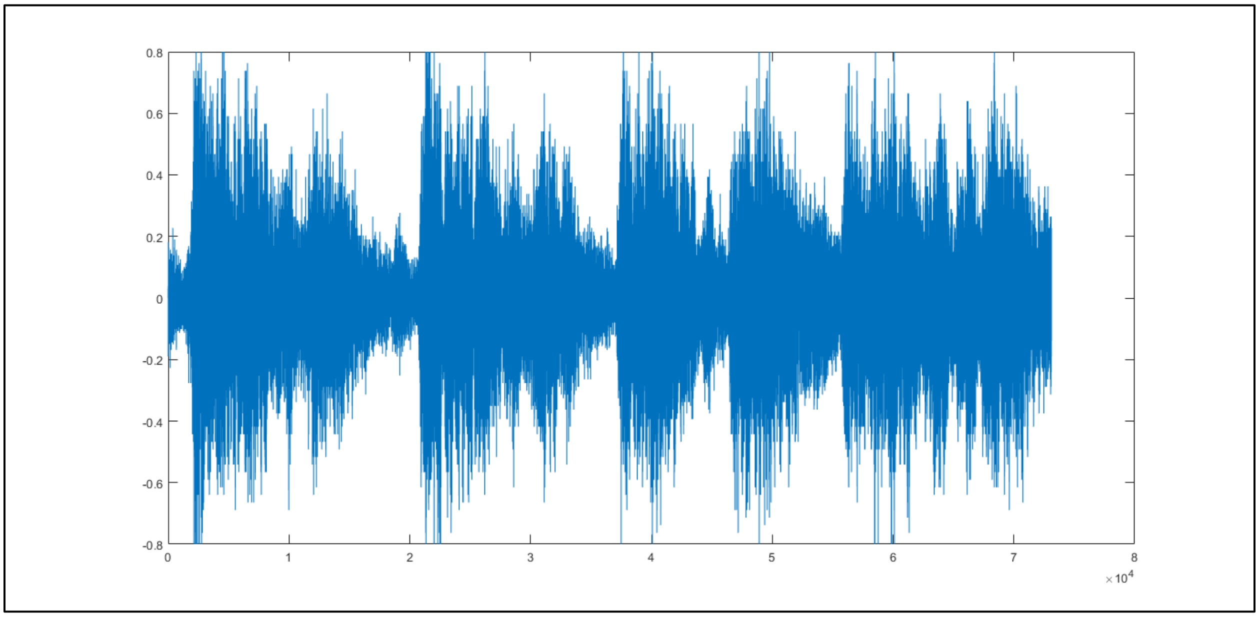


Figure 4.6: Chanting.wav sound file wave

In the lower level (level two DWT steganography), three images with different sizes and dimensions have been used.

The first image was the autumn bench shown in figure 4.7 with dimensions of 3840 x 2160 and used as cover image, the first secret data to be embedded on this cover image is the beeb.wav sound file, the second secret message is the compressed beeb.wav file using the Huffman compression algorithm, the third secret message is the compressed beeb.wav file using the LZW compression algorithm.

The original image is shown on Figure 4.7.

The new stego images are shown in Figure 4.8, Figure 4.9 and Figure 4.10 respectively

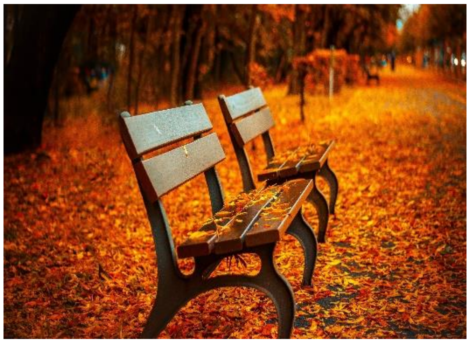


Figure 4.7: Autumn Bench Original Image

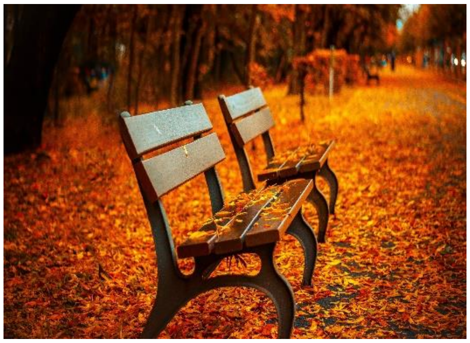


Figure 4.8: Autumn Bench (No Compression)

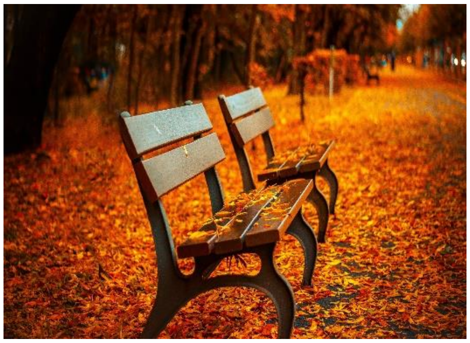


Figure 4.9: Autumn Bench (Huffman Compressed)

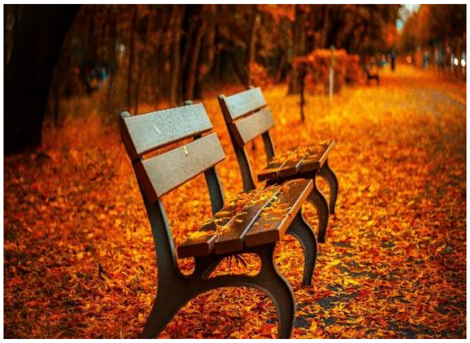
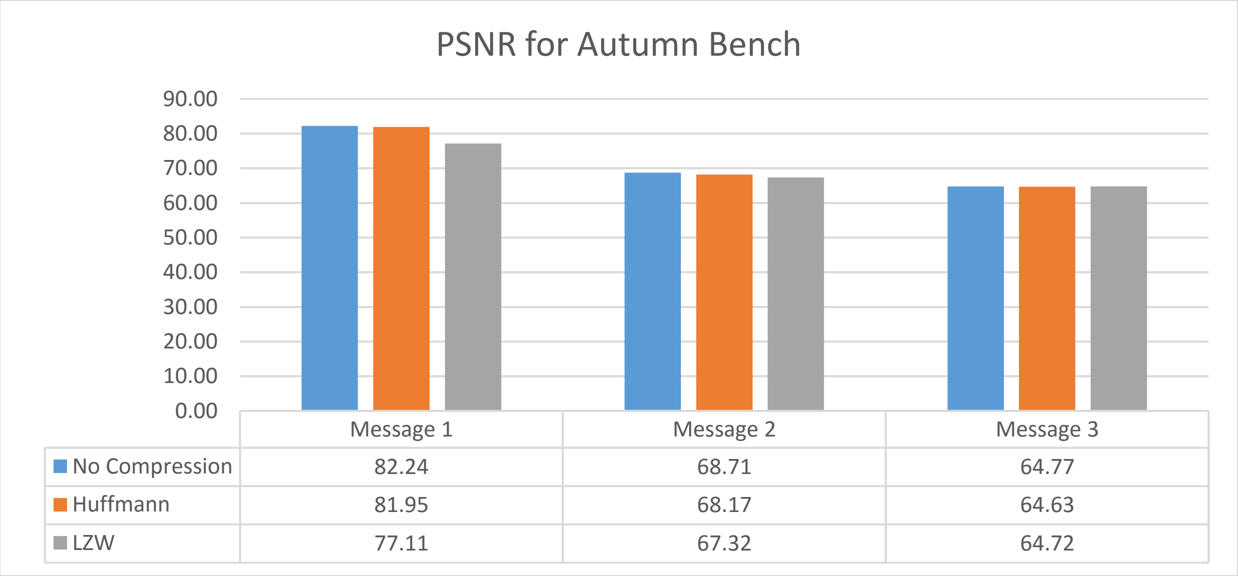


Figure 4.10: Autumn Bench (LZW Compressed)

Table 4.2 below shows the experiment results of the autumn bench stego images with their PSNR and MSE values. Figure 4.11 shows the PSNR diagram of the PSNR values. Figure 4.12 shows MSE diagram of the MSE values.

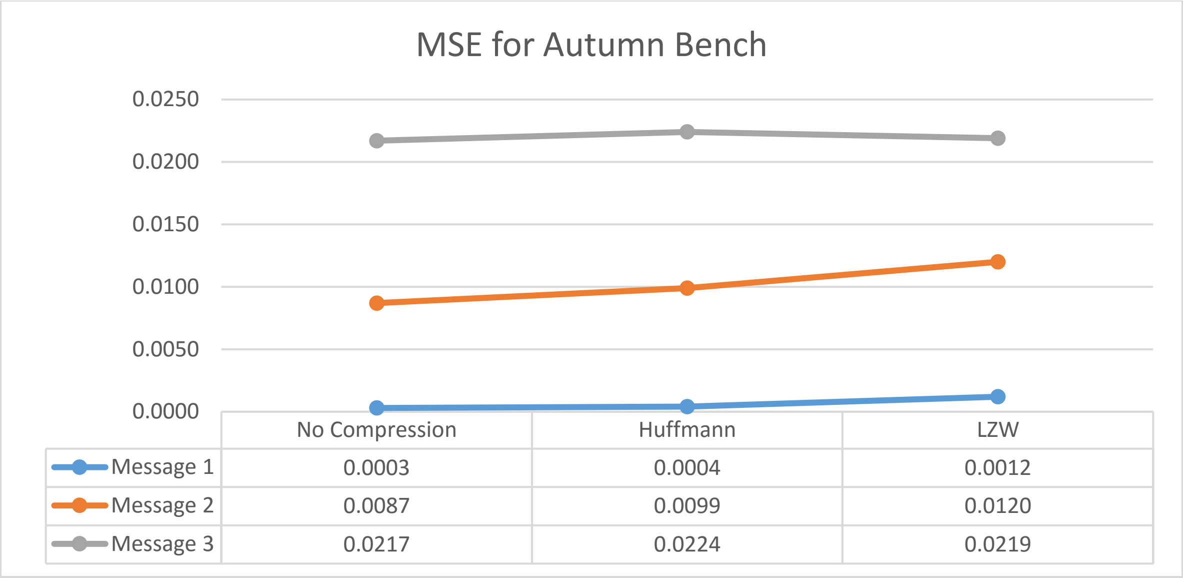
Secret Message	Size of Secret Message in Bits	Cover Audio File (Leve One)	Size of Audio File in Bits	Compression Algorithm	Cover Image (Level Two)	PSNR	MSE
Message 1	7,422	Beep.wav	11,659	No Compression	Autumn Bench (3840 x 2160)	82.23	0.0003
				Huffamn		81.95	0.0004
				LZW		77.10	0.0012
Message 2	163,840	Squeeze.wav	516,096	No Compression	Autumn Bench (3840 x 2160)	68.70	0.0087
				Huffamn		68.16	0.0099
				LZW		67.31	0.0120
Message 3	286,720	Chnating.wav	1,163,264	No Compression	Autumn Bench (3840 x 2160)	67.76	0.0217
				Huffamn		64.62	0.0224
				LZW		64.72	0.0219

Table 4.2 Experimental results of Autumn Bench stego image



	Message 1	Message 2	Message 3
🟦 No Compression	82.24	68.71	64.77
🟧 Huffmann	81.95	68.17	64.63
⬜ LZW	77.11	67.32	64.72

Figure 4.11: PSNR values diagram for Autumn Bench stego images



	No Compression	Huffmann	LZW
Message 1	0.0003	0.0004	0.0012
Message 2	0.0087	0.0099	0.0120
Message 3	0.0217	0.0224	0.0219

Figure 4.12: MSE values diagram for Autumn Bench stego images

The second image used is the Ocean Nature image, with dimensions 3840 x 2160, the first secret data to be embedded on this cover image is the squeeze.wav sound file, the second secret message is the compressed squeeze.wav file using the Huffman compression algorithm, the third secret message is the compressed squeeze.wav file using the LZW compression algorithm.

The original image is shown on Figure 4.13.

The new stego images are shown in Figure 4.14, Figure 4.15 and Figure 4.16 respectively

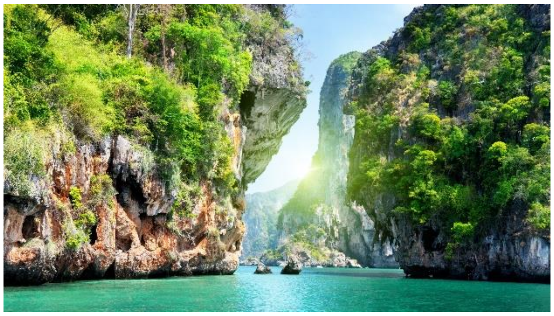


Figure 4.13: Ocean Nature Original Image

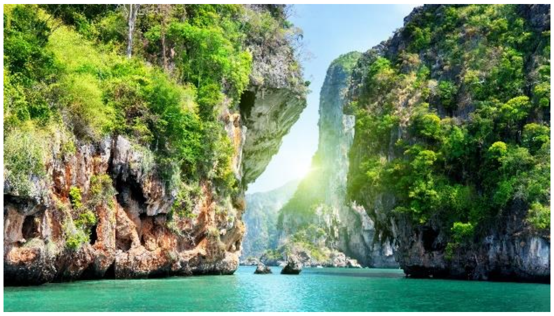


Figure 4.14: Ocean Nature (No Compression)

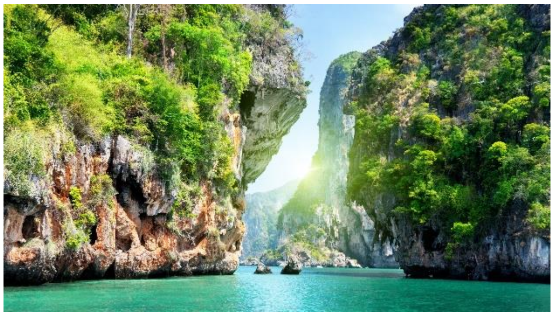


Figure 4.15: Ocean Nature (Huffman Compressed)

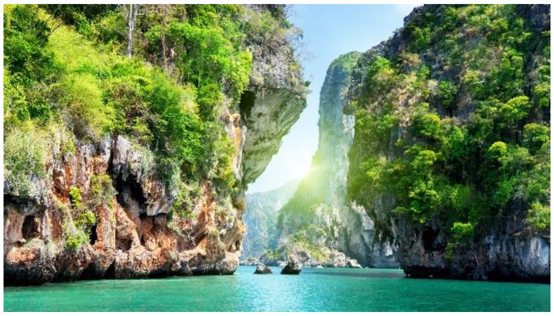
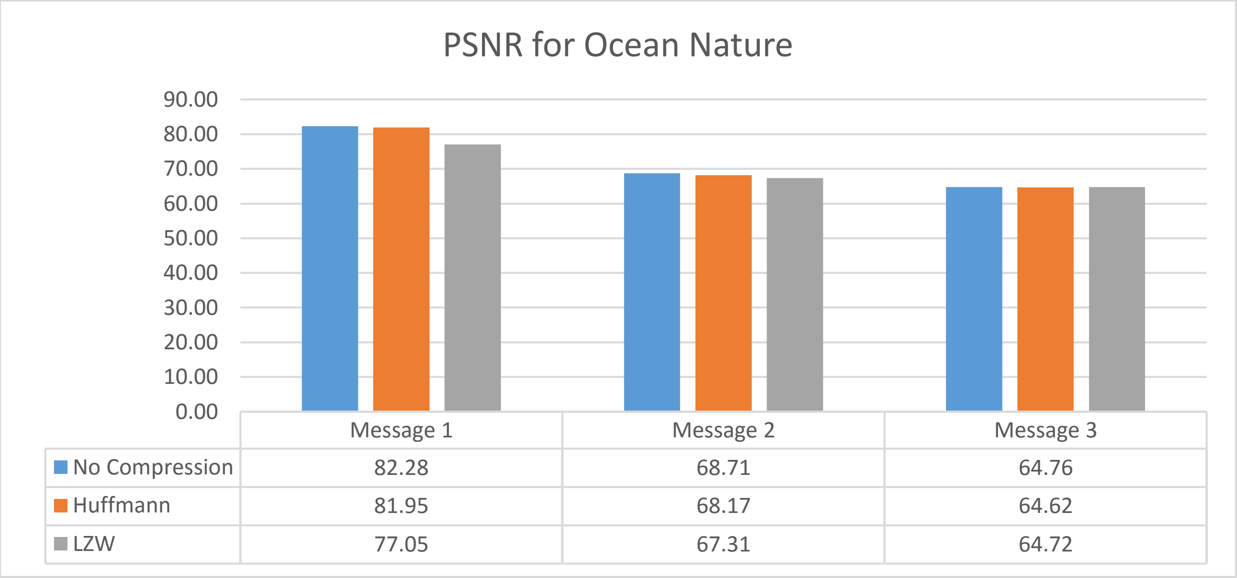


Figure 4.16: Ocean Nature (LZW Compressed)

Table 4.3 below shows the experiment results of the Ocean Nature stego images with their PSNR and MSE values. Figure 4.17 shows the PSNR diagram of the PSNR values. Figure 4.18 shows MSE diagram of the MSE values.

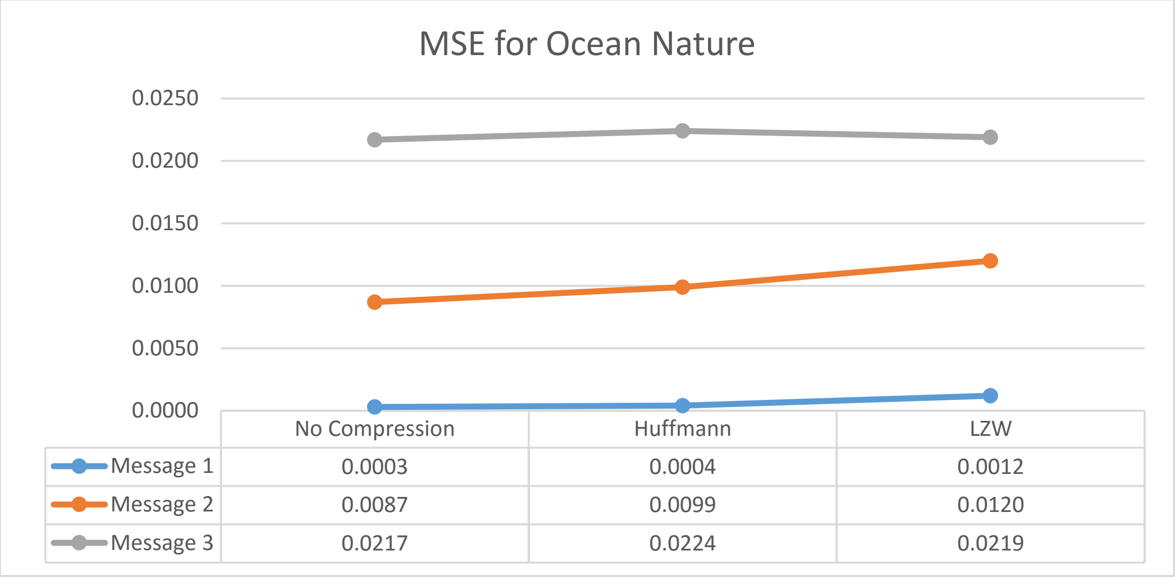
Secret Message	Size of Secret Message in Bits	Cover Audio File (Leve One)	Size of Audio File in Bits	Compression Algorithm	Cover Image (Level Two)	PSNR	MSE
Message 1	7,422	Beep.wav	11,659	No Compression	Ocean Nature (3840 x 2160)	82.28	0.0003
				Huffamn		81.95	0.0004
				LZW		77.05	0.0012
Message 2	163,840	Squeeze.wav	516,096	No Compression	Ocean Nature (3840 x 2160)	68.71	0.0087
				Huffamn		68.17	0.0099
				LZW		67.31	0.0120
Message 3	286,720	Chnating.wav	1,163,264	No Compression	Ocean Nature (3840 x 2160)	64.76	0.0217
				Huffamn		64.62	0.0224
				LZW		64.72	0.0219

Table 4.3 Experimental results of Ocean Nature stego image



	Message 1	Message 2	Message 3
🟦 No Compression	82.28	68.71	64.76
🟧 Huffmann	81.95	68.17	64.62
◼ LZW	77.05	67.31	64.72

Figure 4.17: PSNR values diagram for Ocean Nature stego images



	No Compression	Huffmann	LZW
Message 1	0.0003	0.0004	0.0012
Message 2	0.0087	0.0099	0.0120
Message 3	0.0217	0.0224	0.0219

Figure 4.18: MSE values diagram for Ocean Nature stego images

The third image used is the Hot Balloons image, with dimensions 3840 x 2160, the first secret data to be embedded on this cover image is the chanting.wav sound file, the second secret message is the compressed chanting.wav file using the Huffman compression algorithm, the third secret message is the compressed chanting.wav file using the LZW compression algorithm.

The original image is shown on Figure 4.19.

The new stego images are shown in Figure 4.20, Figure 4.21 and Figure 4.22 respectively

Figure 4.19: Hot Balloons Original Image

Figure 4.20: Hot Balloons (No Compression)

Figure 4.21: Hot Balloons (Huffman Compressed)

Figure 4.22: Hot Balloons (LZW Compressed)

Table 4.4 below shows the experiment results of the Hot Balloons stego images with their PSNR and MSE values. Figure 4.23 shows the PSNR diagram of the PSNR values. Figure 4.24 shows MSE diagram of the MSE values.

Secret Message	Size of Secret Message in Bits	Cover Audio File (Leve One)	Size of Audio File in Bits	Compression Algorithm	Cover Image (Level Two)	PSNR	MSE
Message 1	7,422	Beep.wav	11,659	No Compression	Hot Balloons (3840 x 2160)	82.24	0.0003
				Huffamn		81.94	0.0004
				LZW		77.09	0.0012
Message 2	163,840	Squeeze.wav	516,096	No Compression	Hot Balloons (3840 x 2160)	68.71	0.0087
				Huffamn		68.16	0.0099
				LZW		67.32	0.0120
Message 3	286,720	Chnating.wav	1,163,264	No Compression	Hot Balloons (3840 x 2160)	64.77	0.0217
				Huffamn		64.64	0.0223
				LZW		64.72	0.0219

Table 4.4 Experimental results of Hot Balloons stego image

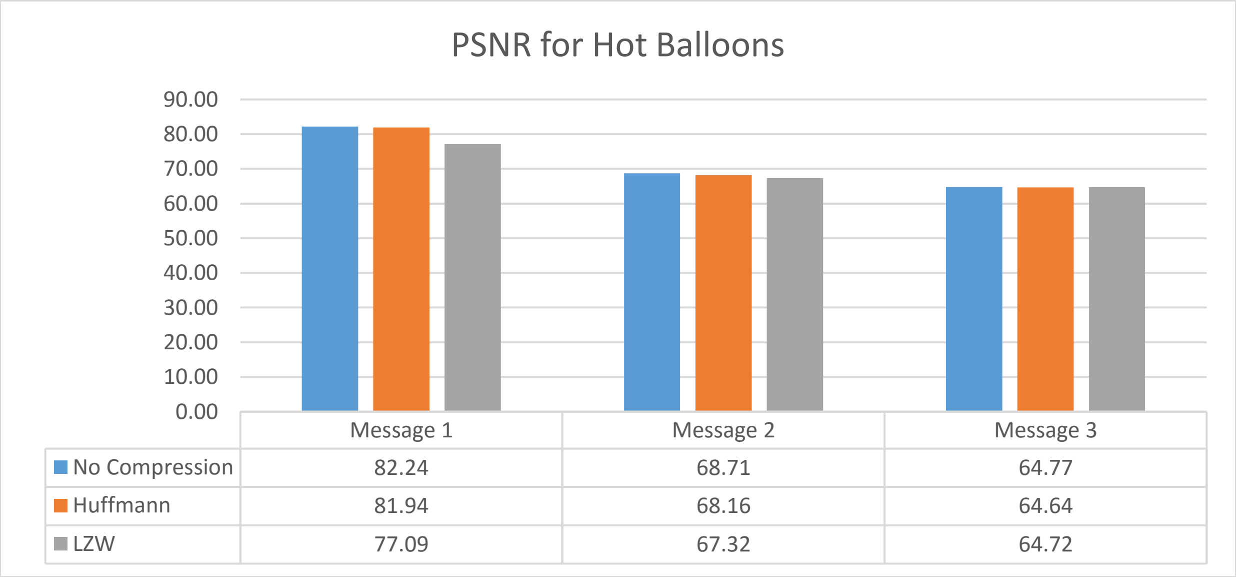


Figure 4.23: PSNR values diagram for Hot Balloons stego images

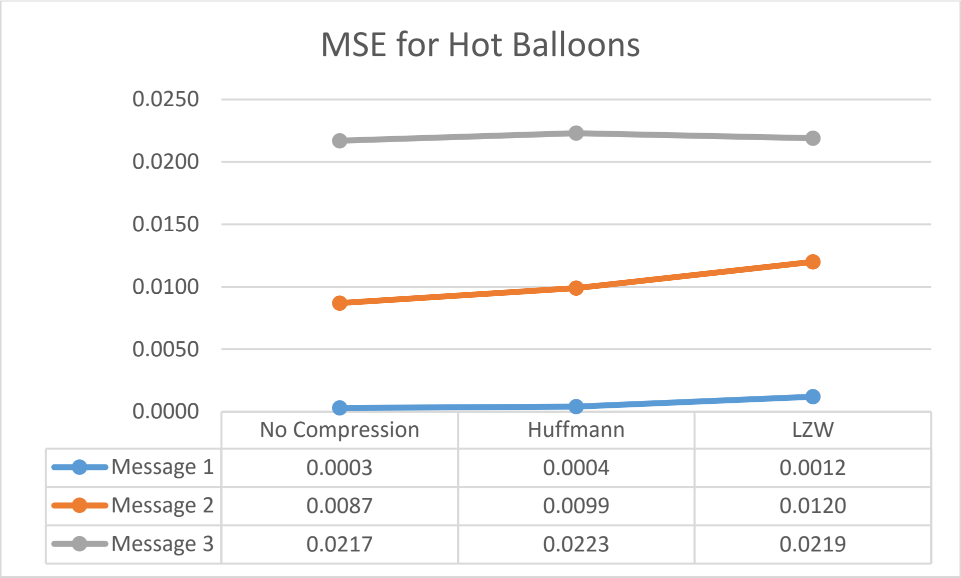


Figure 4.24: MSE values diagram for Hot Balloons stego images

4.5 Discussion

Experiments were done on cover audio files with different sizes and cover images with different sizes as well in both stages, the secret messages were also different sizes and hidden inside the cover audio files. Lossless compression algorithms were used on cover audio files. MSE and PSNR were calculated for each cover image and the following results came out:

1. PSNR gives better results when the cover audio file is not compressed.
2. The two compression algorithms don't give a major effect on the steganography process, the PSNR values are close in cases the compression is used or not.
3. PSNR values of Huffman compression were better than PSNR values of LZW compression.
4. The length of the text files used in the first steganography process don't have major effect on the second steganography process.

CHAPTER V
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Steganography is the art and science of hiding secret information in a cover file such that only sender and receiver can detect the existence of the secret information, Steganography can be classified into image, text, audio and video, based on the cover media used to embed secret data. The goal of steganography to hide message in such a way that no one apart from the intended recipient even know that the massage has been sent.

The main objective is applying the lossless compression algorithms and study the effects of compression on audio steganography. In this thesis, Multi-Level Steganography for audio steganography, was presented. MLS consists of at least two stenographic methods are utilized respectively, in such a way that one method (called the upper-level) as a carrier for the second one (called the lower-level). The first level has been applied using enhanced random LSB audio steganography and the second level applied using DWT image steganography, then applying compression process between first and second level of steganography. For audio compression two types of lossless compression have been applied, Huffman coding and LZW coding.

The system is better when the audio file is not compressed because the quality of original data and stego date are the same but it take a lot of unnecessary space to store. The system used lossless compression to keep the audio quality of the original source using a less space.

The Huffman algorithm was able to reduce the data size by 43% on average, which is four times better than the LZW algorithm.

5.2 Recommendations

1. Apply encryption to the secret message before level one steganography.

2. Use another compression algorithms.

3. Add additional encryption layer to encrypt the compressed audio before performing level two steganography.

5.3 Future Work

- Add additional functions to the system, like watermarking algorithm, encryption algorithm.

- Add support to Video and other audio formats.

REFERENCES

[1] Dr. Atef Jawad Al-Najar, “Multi-Level Digital Multimedia Steganography Model” Heraklion, Greece, 2008.

[2] Aiswarya T, Mansi Shah, Aishwarya Talekar, Pallavi Raut “Steganographic Technique for Hiding Secret Audio in an Image”, Mumbai, India, IJREAM, 2017, Vol. 03, Issue 04.

[3] Vijaya Lakshmi Chittimalli “Implementation Of Steganography With Audio File And Encrypted Document”, California, USA, California State University, 2009.

[4] Suresh Babu. P. “Secure Data Communication using Steganography and Cryptography”, Mumbai, India, IJARCCE 2014.

[5] Mazdak Zamani , Hamed Taherdoost, Azizah A. Manaf , Rabiah B. Ahmad , and Akram M. Zeki, “Robust Audio Steganography via Genetic Algorithm”, Karachi, Pakistan, IEEE 2009.

[6] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn “Information Hiding - A Survey”, IEEE,1999.

[7] Jayaram P, Ranganatha H, Anupama H, "Information hiding using audio steganography - A Survey", Bangalore, India, IIJMA, 2011.

[8] Currie, D.L. & Irvine, C.E., “Surmounting the effects of lossy compression on Steganography”, California, USA, Naval Postgraduate School, 1996.

[9] Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, in IEEE Journal of selected Areas in Communications, Vol.16, Issue. 04, 1998.

[10] Tamanna Garg, Sonia Vatta “Research paper on enhancing data compression rate using steganography”, Bahara, India, IJCMS, Vol. 03, Issue. 04, 2014.

[11] Methods of Audio Steganography, Mumbai, India, 2010.

.

[12] Chintan R. Nagrecha , Prof. Prashant B. Swadas “Audio Steganography with Various Compression Algorithms to Improve Robustness and Capacity”, India, IJARCSSE, Vol. 04, Issue. 05, 2014.

[13] K.P.Adhiya,K.P.Adhiya Swati A. Patil,"Hiding Text in Audio Using LSB Based Steganography", Bambhori, India, IISTE, Vol. 02, Issue. 03, 2011

[14] M.Baritha Beguma ,Y.Venkataramanib "LSB Based Audio Steganography Based On Text Compression", India, ELSEVIER, 2011.

[15] Pratishtha Gupta1, G.N Purohit2, Varsha Bansal3 "A Survey on Image Compression Techniques" Rajasthan, India, IJARCCE, Vol. 03, Issue. 08, 2014.

[16] AmbiKadev, LijilDomininc, Swetha "Compress-Encrypt Video Steganography", India, IJIRST, Vol. 01, Issue. 11, 2015.

[17] Rahul Jain, Naresh Kumar "Efficient data hiding scheme using lossless data compression and image steganography", Kurukshetra, India, IJEST, 2012.

[18] Suma Christal Mary presented "Improved protection in video steganography used compressed video bitstreams", Tamilnadu, India, IJCSE, Vol. 02, Issue. 03, 2010.